



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/847,067	04/30/2001	Brian T. Murren	GE1-0005US	4549
21718	7590	07/29/2010	EXAMINER	
LEE & HAYES PLLC			DESAI, RACHNA SINGH	
601 W. RIVERSIDE AVENUE			ART UNIT	PAPER NUMBER
SUITE 1400			2176	
SPOKANE, WA 99201				
NOTIFICATION DATE		DELIVERY MODE		
07/29/2010		ELECTRONIC		

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

lhpto@lechayes.com

Office Action Summary	Application No. 09/847,067	Applicant(s) MURREN ET AL.
	Examiner RACHNA S. DESAI	Art Unit 2176

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If no period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).

Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(o).

Status

1) Responsive to communication(s) filed on 15 March 2010.

2a) This action is FINAL. 2b) This action is non-final.

3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) Claim(s) _____ is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5) Claim(s) _____ is/are allowed.

6) Claim(s) _____ is/are rejected.

7) Claim(s) _____ is/are objected to.

8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) The specification is objected to by the Examiner.

10) The drawing(s) filed on _____ is/are: a) accepted or b) objected to by the Examiner.
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a) All b) Some * c) None of:

1. Certified copies of the priority documents have been received.
2. Certified copies of the priority documents have been received in Application No. _____.
3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) Notice of References Cited (PTO-892)
- 2) Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) Information Disclosure Statement(s) (PTO/SB/08)
 Paper No(s)/Mail Date _____
- 4) Interview Summary (PTO-413)
 Paper No(s)/Mail Date _____
- 5) Notice of Informal Patent Application
- 6) Other: _____

DETAILED ACTION

1. This action is responsive to communications: A request for continued examination under 37 CFR 1.114 was filed in this application after a decision by the Board of Patent Appeals and Interferences, but before the filing of a Notice of Appeal to the Court of Appeals for the Federal Circuit or the commencement of a civil action. Since this application is eligible for continued examination under 37 CFR 1.114 and the fee set forth in 37 CFR 1.17(e) has been timely paid, the appeal has been withdrawn pursuant to 37 CFR 1.114 and prosecution in this application has been reopened pursuant to 37 CFR 1.114. Applicant's submission filed on 03/15/10 has been entered.

2. Claims 1-2, 4-15, 17-19, 30-34, and 37-39 are pending.

Specification

3. The specification is objected to as failing to provide proper antecedent basis for the claimed subject matter. See 37 CFR 1.75(d)(1) and MPEP § 608.01(o). Correction of the following is required: "computer readable media" has not been defined in the Specification.

Claim Rejections - 35 USC § 102

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

5. Claims 1-2, 4-10, and 37 are rejected under the principles of Res judicata.

Claims 1-2, 4-10, and 37 are unchanged from the decision by the Board of Patent Appeals and Interferences, mailed 01/15/2010, affirming the rejection of claims 1-2, 4-10, and 37 over the Kougiouris reference. Claim 1, although amended, is not patentably distinct from claim 1 as affirmed by the Board of Patent Appeals and Interferences. The sole change to claim 1 is removing the term "selected" from the limitation, "selected ones of the set of one or more attributes". As it was shown in the examiner's answer to appeal mailed 05/01/2008 that Kougiouris teaches creating code for one or more forms including ones of the set of one or more attributes, the claim is not patentably distinct from that affirmed by the Board of Patent Appeals and Interferences.

6. Claims 1-2, 4-10, and 37 are rejected under 35 U.S.C. 102(e) as being anticipated by Kougiouris et al., US 2004/0039993 A1, 2/26/04 (Filed 8/27/03, Continuation of application filed 11/15/99).

Kougiouris was cited as a prior art reference in the Examiner's Answer to appeal, mailed 05/01/2008.

7. The rejections of claims 1-2, 4-10, and 37 are respectfully maintained and outlined below. Applicant having made only a minor non-distinct change to claim 1, these claims remain anticipated by Kougiouris as decided by the decision by the Board of Patent Appeals and Interferences.

8. **Claims 1-2, 4-10, 37 and 39 are rejected under 35 U.S.C. 102(e) as being anticipated by Kougiouris et al., US 2004/0039993 A1, 2/26/04 (Filed 8/27/03, Continuation of application filed 11/15/99).**

In reference to claim 1, Kougiouris teaches an automatic formatting and validating of text for markup language graphical user interface (GUI). The GUI markup language description comprises various types of GUI elements for which text is to be validated and formatted such as form fields, tables, and links. See page 1, paragraph [0010]. The GUI element may comprise one or more fields for accepting text input and displaying text output. The markup language file GUI descriptions comprise information usable by the validation/formatting manager component to perform various types of

validating/formatting operations. This information may exist as markup language tag attributes, e.g., by adding custom attributes to markup language. See page 4, paragraphs [0061]-[0064]. The user may provide text input to the GUI element which is validated by the manager before it is displayed in HTML form. See page 5, paragraph [0070]-[0075]. This meets the limitations, ***accessing a computer program; automatically identifying a set of one or more attributes of the computer program with values that are to be input to the computer program by a user.*** Kougiouris further discloses that Graphical user interfaces (GUIs) often include text fields for accepting text input or displaying text output. For example, graphical user interfaces may comprise a "form", that is a series of text fields with a look and feel similar to a paper-based form. Many text fields are designed to accept text input or display text output that is often formatted or demarcated in a particular way. See page 1, paragraphs [0005]. The user may provide text input to the GUI element which is validated by the manager before it is displayed in HTML form. See page 5, paragraph [0070]-[0075]. The graphical user interfaces can be created from markup languages such as HTML or XML-derived markup language descriptions which meets the limitation, ***creating code for one or more forms including ones of the set of one or more attributes.***

In reference to claim 2, Kougiouris discloses outputting the attributes in a form such as an HTML form in which the various attributes are listed. See figures 5A-5C.

In reference to claims 4-5, Kougiouris discloses that Graphical user interfaces (GUIs) often include text fields for accepting text input or displaying text output. For example, graphical user interfaces may comprise a "form", that is a series of text fields with a look and feel similar to a paper-based form. Many text fields are designed to accept text input or display text output that is often formatted or demarcated in a particular way. See page 1, paragraphs [0005]. The user may provide text input to the GUI element which is validated by the manager before it is displayed in HTML form. See page 5, paragraph [0070]-[0075]. The graphical user interfaces can be created from markup languages such as HTML or XML-derived markup language descriptions.

In reference to claim 6, Kougiouris discloses the user may provide text input to the GUI element which is validated by the manager before it is displayed in HTML form. See page 5, paragraph [0070]-[0075]. See also figures 5A-5C which illustrate a data input field for inputting a value for the attributes. The user may also perform various other actions causing the application to check the text, such as issuing a command to submit the data the user has entered to a database, or perform other types of transactions using the data. See page 8, paragraph [0125].

In reference to claim 7, Kougiouris teaches the user may perform various other actions causing the application to check the text, such as issuing a command to submit the data the user has entered to a database, or perform other types of transactions using the data. See page 8, paragraph [0125].

In reference to claim 8, Kougouris discloses the information may exist as markup language tag attributes, e.g., by adding custom attributes to markup language. See page 4, paragraphs [0061]-[0064]. The user may provide text input to the GUI element which is validated by the manager before it is displayed in HTML form. See page 5, paragraph [0070]-[0075].

In reference to claim 9, Kougouris discloses outputting the attributes in a form such as an HTML form in which the various attributes are listed. See figures 5A-5C.

In reference to claim 10, Kougouris discloses that Graphical user interfaces (GUIs) often include text fields for accepting text input or displaying text output. For example, graphical user interfaces may comprise a "form", that is a series of text fields with a look and feel similar to a paper-based form. Many text fields are designed to accept text input or display text output that is often formatted or demarcated in a particular way. See page 1, paragraphs [0005]. The user may provide text input to the GUI element which is validated by the manager before it is displayed in HTML form. See page 5, paragraph [0070]-[0075]. The graphical user interfaces can be created from markup languages such as HTML or XML-derived markup language descriptions.

Regarding claim 37, Kougiouris teaches presenting a GUI with a form including text fields for accepting text input. The GUI may comprise a form that is a series of text fields with a look and feel similar to a paper-based form. See page 1, paragraph [0005].

Regarding claim 39, Kougiouris teaches an automatic formatting and validating of text for markup language graphical user interface (GUI). The GUI markup language description comprises various types of GUI elements for which text is to be validated and formatted such as form fields, tables, and links. See page 1, paragraph [0010]. The GUI element may comprise one or more fields for accepting text input and displaying text output. The markup language file GUI descriptions comprise information usable by the validation/formatting manager component to perform various types of validating/formatting operations. This information may exist as markup language tag attributes, e.g., by adding custom attributes to markup language. See page 4, paragraphs [0061]-[0064]. The user may provide text input to the GUI element which is validated by the manager before it is displayed in HTML form. See page 5, paragraph [0070]-[0075]. This meets the limitations, ***wherein the creating code for the one or more forms is performed automatically and is based on the automatic identifying of the set of the one or more attributes.***

9. Claims 30-32 and 34 are rejected under 35 U.S.C. 102(e) as being anticipated by Cifra et al., US 7,376,904 B2, 05/20/08 (11/29/00).

Regarding claim 30, Cifra discloses a program information comprises prototype information which specifies a plurality of functional operations, wherein each functional operation has associated input and output parameters which meets the limitation, ***accessing the computer program to identify operations in the computer program that load attribute values and set attribute values.*** See column 13, lines 4-14.

Cifra teaches ***analyzing the identified operations to determine one or more user inputs to the computer program;*** See column 3, lines 36-50, column 4, lines 3-9 and 21-40, and column 14, lines 29-51 and 62-67 and column 15, lines 41-52 which discloses identifying attributes of a computer program and outputting the input parameters in a GUI.

Cifra discloses generating a GUI to allow a user to input some of the user inputs which meets the limitation, ***automatically generating code for one or more input forms to allow a user to input at least some of the one or more user inputs.*** See column 3, lines 36-50 and column 4, lines 3-9, 21-40 and 54-67.

Cifra discloses a method and system for creating a GUI or a generated program, such that users can specify input parameter values via user interface controls in the GUI and may view output parameter values via user interface controls. Specifically, a computer program is generated including a GUI for specifying input values to or viewing output values from the program which meets the limitation, ***accessing the computer program to identify one or more outputs of the computer program and automatically generating code for one or more output forms to present the outputs of the computer program.*** See column 3, lines 35-50, column 4, lines 3-9

and 21-40 which discloses a GUI for viewing output parameter values from the program. See also column 4, lines 64-67 through column 5, lines 1-9 which discloses displaying a list of the output parameters. See also column 14, lines 52-67.

Regarding claim 31, Cifra discloses *generating a list identifying the one or more user inputs to the computer program and another list identifying the one or more outputs of the computer program; and output the lists.* See column 3, lines 35-50 which discloses creating a GUI for a generated program from which a user can view output parameter values. See also column 4, lines 3-9 and 21-40 which discloses a GUI for viewing output parameter values from the program. See also column 4, lines 64-67 through column 5, lines 1-9 which discloses displaying a list of the output parameters. See also column 14, lines 52-67. See column 3, lines 36-50, column 4, lines 3-9 and 21-40, and column 14, lines 29-51 and 62-67 and column 15, lines 41-52 which discloses identifying attributes of a computer program and outputting the input parameters in a GUI.

Regarding claim 32, Cifra discloses *identifying the selected one or more user inputs to include on the form; creating data input fields for the form definition via which a user can subsequently input a value for the user inputs; and creating a submit tag for the form definition via which the user can subsequently input a request to submit the values on the form to the computer program.* See column 3, lines 36-50, column 4, lines 3-9 and 21-40, and column 14,

lines 29-51 and 62-67 and column 15, lines 41-52 which discloses identifying attributes of a computer program and outputting the input parameters in a GUI. See column 4, lines 21-40 which discloses a user can adjust the parameter value as desired, e.g. in order to experimentally see the effect of varying the parameter value. The GUI allows a user to set input parameter values. See also column 4, lines 54-67 which discloses a list of input parameters can be interactively changeable through a GUI. See column 5, lines 10-20 which discloses a user interface control associated with the item may include a field for setting each value. See figure 18 which illustrates a GUI for specifying input parameters and a "finish" button which submits the values to the computer program.

Regarding claim 34, Cifra teaches the attributes of the computer program can be identified which includes all attributes of the program which meets the limitation, *identify one or more attributes of the computer program that are not obtained by the computer program from elsewhere and that cannot be user inputs.* See column 4, lines 4-67.

Claim Rejections - 35 USC § 103

10. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the

invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

11. Claims 14-15, 17, and 38 are rejected under 35 U.S.C. 103(a) as being unpatentable over Cifra et al., US 7,376,904 B2, 05/20/08 (11/29/00) in view of Rumbaugh et al., US 5,005,119, 04/02/91.

Regarding claim 14, Cifra discloses *accessing a computer program; automatically identifying a set of one or more outputs of the computer program; generating a list identifying the set of one or more outputs; and outputting the list.* See column 3, lines 35-50 which discloses creating a GUI for a generated program from which a user can view output parameter values. See also column 4, lines 3-9 and 21-40 which discloses a GUI for viewing output parameter values from the program. See also column 4, lines 64-67 through column 5, lines 1-9 which discloses displaying a list of the output parameters. See also column 14, lines 52-67.

Cifra does not teach the identifying and generating are performed based on an analysis of the computer program, independent of the execution of the computer program.

Rumbaugh discloses *wherein the identifying and generating are performed based on an analysis of the computer program code, independent of execution of the computer program to provide one or more views.* See column 9, lines 7-26 which discloses identifying and generating a list of inputs and outputs without execution of the program.

It would have been obvious to a person of ordinary skill in the art at the time of the invention to have incorporated Rumbaugh's identification and generation output lists of the program without executing the program within the system of Cifra because it was often desirable to know the output data sets of a computer program for use as input in another program without having to actually execute all programs.

Regarding claim 15, Cifra discloses ***automatically identifying a set of one or more attributes of the computer program with values that are to be input to the computer program by a user; and outputting an identification of the set of one or more attributes.*** See column 3, lines 36-50, column 4, lines 3-9 and 21-40, and column 14, lines 29-51 and 62-67 and column 15, lines 41-52 which discloses identifying attributes of a computer program and outputting the input parameters in a GUI.

Regarding claim 17, Cifra discloses ***creating one or more forms including selected ones of the set of one or more outputs.*** See column 3, lines 35-50, column 4, lines 3-9 and 21-40 which discloses a GUI for viewing output parameter values from the program. See also column 4, lines 64-67 through column 5, lines 1-9 which discloses displaying a list of the output parameters. See also column 14, lines 52-67.

Regarding claim 38, Cifra discloses ***generating code for one or more input forms and automatically generating code for said one or more output forms are***

performed in a form generation procedure in which said one or more input forms and said one or more output forms are developed. See column 3, lines 35-50 which discloses creating a GUI for a generated program from which a user can view input and output parameter values. See also column 4, lines 3-9 and 21-40 which discloses a GUI for viewing input and output parameter values from the program. See also column 4, lines 64-67 through column 5, lines 1-9 which discloses displaying a list of the input and output parameters. See also column 14, lines 52-67.

Cifra does not teach the identifying and generating are performed based on an analysis of the computer program, independent of the execution of the computer program.

Rumbaugh discloses ***the form generation procedure being independent and antecedent to a user's interaction with the computer program via said one or more input forms and said one or more output forms.*** See column 9, lines 7-26 which discloses identifying and generating a list of inputs and outputs without execution of the program.

It would have been obvious to a person of ordinary skill in the art at the time of the invention to have incorporated Rumbaugh's identification and generation output lists of the program without executing the program within the system of Cifra because it was often desirable to know the output data sets of a computer program for use as input in another program without having to actually execute all programs.

12. Claims 18-19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Cifra et al., US 7,376,904 B2, 05/20/08 (11/29/00) in view of Rumbaugh et al., US 5,005,119, 04/02/91, as applied to claim 14, and further in view of Gloudeman et al., US 6,028,998, 02/22/00 (filed 04/03/98).

Regarding claim 18, Cifra discloses identifying the outputs included in a computer program which meets the portion of the limitation, ***identifying, as one of the set of one or more outputs, each output included in one of the identified methods.*** See column 3, lines 35-50, column 4, lines 3-9 and 21-40 which discloses a GUI for viewing output parameter values from the program. See also column 4, lines 64-67 through column 5, lines 1-9 which discloses displaying a list of the output parameters. See also column 14, lines 52-67.

Cifra does not expressly disclose a view definition.

However, Gloudeman discloses ***the computer program includes a plurality of interactions that each include one or more view definitions, wherein each view definition defines a view that is a response to a request and identifying, for each of the view definitions of each of the plurality of interactions, the methods of the view definition and identifying, each output included in one of the identified methods.*** See column 4, lines 7-21 which discloses a view component that stores methods defined for displaying different sets of predefined attributes for providing information to a user. See column 10, lines 20-24 which discloses a view definition which includes an attribute list for each view.

It would have been obvious to a person of ordinary skill in the art at the time of the invention to have included Gloudeman's view definitions as part of the computer program of Cifra since computer programs can include any methods.

Regarding claim 19, Cifra discloses identifying the outputs included in a computer program which meets the portion of the limitation, *creating a form for each of the plurality of interactions, and wherein each form includes one or more outputs for the corresponding one of the plurality of interactions.* See column 3, lines 35-50, column 4, lines 3-9 and 21-40 which discloses a GUI for viewing output parameter values from the program. See also column 4, lines 64-67 through column 5, lines 1-9 which discloses displaying a list of the output parameters. See also column 14, lines 52-67.

Cifra does not disclose a plurality of interactions including command definitions and view definitions. However, Gloudeman discloses command definitions and view definitions. See column 4, lines 7-21 which discloses a view component that stores methods defined for displaying different sets of predefined attributes for providing information to a user. See column 10, lines 20-24 which discloses a view definition which includes an attribute list for each view. See also column 3, lines 51-56 and column 4, lines 22-38 which discloses a command component which are methods that are visible to outside objects and to the user interface. The command definition includes variables for each command. These teachings meet the limitations, *a plurality of interactions that each include one or more command definitions and one or*

more view definitions, wherein each command definition defines a command having various attributes and a behavior, wherein each view definition defines a view that is a response to a request.

It would have been obvious to a person of ordinary skill in the art at the time of the invention to have included Gloudeman's view and command definitions as part of the computer program of Cifra since computer programs can include any methods.

13. Claims 11, 13, and 33 are rejected under 35 U.S.C. 103(a) as being unpatentable over Cifra et al., US 7,376,904 B2, 05/20/08 (11/29/00) in view of Gloudeman et al., US 6,028,998, 02/22/00 (filed 04/03/98).

Regarding claim 11, Cifra discloses program information comprises prototype information which specifies a plurality of functional operations, wherein each functional operation has associated input and output parameters where the parameters may be set by an operation or provided to the operation which meets the limitation, ***accessing a computer program to identify operations in the computer program that load attribute values and set attribute values.*** See column 13, lines 4-14.

Cifra teaches ***analyzing the identified operations; identifying . . . the methods; checking, for each identified method that sets a value, whether a corresponding identified method obtains a value, and identifying, as an attribute of the set of one or more attributes, each attribute corresponding to a method that sets a value for the attribute for which there is no corresponding identified***

method that obtains the value for the attribute; and outputting an identification of the set of one or more attributes. See column 4, lines 3-40, and 54-67 which discloses functional operations have parameters associated with the operation and displaying a list of parameters associated with the operations. These parameters may be set by the operations or obtained by the operation. See also column 3, lines 36-50, column 4, lines 3-9 and 21-40, and column 14, lines 29-51 and 62-67 and column 15, lines 41-52 which discloses identifying attributes of a computer program and outputting the input parameters in a GUI.

Cifra does not disclose a plurality of interactions including command definitions and view definitions or identifying the methods of the command definition.

However, Gloudeman discloses command definitions and view definitions. See column 4, lines 7-21 which discloses a view component that stores methods defined for displaying different sets of predefined attributes for providing information to a user. See column 10, lines 20-24 which discloses a view definition which includes an attribute list for each view. See also column 3, lines 51-56 and column 4, lines 22-38 which discloses a command component which are methods that are visible to outside objects and to the user interface. The command definition includes variables for each command. These teachings meet the limitations, ***a plurality of interactions that each include one or more command definitions and one or more view definitions, wherein each command definition defines a command having various attributes and a behavior, wherein each view definition defines a view that is a response to a request.***

It would have been obvious to a person of ordinary skill in the art at the time of the invention to have included Gloudeman's view and command definitions as part of the computer program of Cifra since computer programs can include any methods.

Regarding claim 13, Cifra teaches the attributes of the computer program can be identified which includes all attributes of the program which meets the limitation, *identify one or more attributes of the computer program that are not obtained by the computer program from elsewhere and that cannot be user inputs.* See column 4, lines 4-67.

Regarding claim 33, Cifra discloses program information comprises prototype information which specifies a plurality of functional operations, wherein each functional operation has associated input and output parameters where the parameters may be set by an operation or provided to the operation which meets the limitation, *accessing the computer program to identify operations in the computer program that load attribute values and set attribute values.* See column 13, lines 4-14.

Cifra teaches *analyzing the identified operations; identifying. . .the methods; checking, for each identified method that sets a value, whether a corresponding identified method obtains a value, and identifying, as one or more user inputs, each attribute corresponding to a method that sets a value for the attribute for which there is no corresponding identified method that obtains the value for the attribute.* See column 4, lines 3-40, and 54-67 which discloses functional

Art Unit: 2176

operations have parameters associated with the operation and displaying a list of parameters associated with the operations. These parameters may be set by the operations or obtained by the operation. See also column 3, lines 36-50, column 4, lines 3-9 and 21-40, and column 14, lines 29-51 and 62-67 and column 15, lines 41-52 which discloses identifying attributes of a computer program and outputting the input parameters in a GUI.

Cifra does not disclose a plurality of interactions including command definitions and view definitions or identifying the methods of the command definition.

However, Gloudeman discloses command definitions and view definitions. See column 4, lines 7-21 which discloses a view component that stores methods defined for displaying different sets of predefined attributes for providing information to a user. See column 10, lines 20-24 which discloses a view definition which includes an attribute list for each view. See also column 3, lines 51-56 and column 4, lines 22-38 which discloses a command component which are methods that are visible to outside objects and to the user interface. The command definition includes variables for each command. These teachings meet the limitations, ***a plurality of interactions that each include one or more command definitions and one or more view definitions, wherein each command definition defines a command having various attributes and a behavior, wherein each view definition defines a view that is a response to a request.***

It would have been obvious to a person of ordinary skill in the art at the time of the invention to have included Gloudeman's view and command definitions as part of the computer program of Cifra since computer programs can include any methods.

14. Claim 12 is rejected under 35 U.S.C. 103(a) as being unpatentable over Cifra et al., US 7,376,904 B2, 05/20/08 (11/29/00) in view of Gloudeman et al., US 6,028,998, 02/22/00 (filed 04/03/98), as applied to claim 11, and further in view of Ionescu, US 7,150,007 B2, 12/12/06 (filed 05/30/01, provisional filed 09/11/00).

Regarding claim 12, Cifra/Gloudeman disclose command definitions as outlined in claim 11 above; however, they not teach identifying the methods of the command definition comprises querying the command definition to cause the command definition to identify its own methods. However, Ionescu teaches identifying methods of an object based on the definition of the object which meets the portion of the limitation, ***querying the definition to cause the definition to identify its own methods.*** See abstract, column 1, lines 46-67 and column 2, lines 1-7. See also claim 1.

It would have been obvious to a person of ordinary skill in the art at the time of the invention to include Ionescu's querying a definition to identify its methods in Cifra/Gloudeman with because it facilitates the testing of objects in a program without requiring a user to write code to test each object. See column 1, lines 45-55.

Response to Arguments

15. Applicant's arguments with respect to claims 11-19 and 30-39 have been considered but are moot in view of the new ground(s) of rejection.

Claims 1-2, 4-10, and 37 are unchanged from the decision by the Board of Patent Appeals and Interferences, mailed 01/15/2010, affirming the rejection of claims 1-2, 4-10, and 37 over the Kougouris reference. Claim 1, although amended, is not patentably distinct from claim 1 as affirmed by the Board of Patent Appeals and Interferences. The sole change to claim 1 is removing the term "selected" from the limitation, "selected ones of the set of one or more attributes". As it was shown in the examiner's answer to appeal mailed 05/01/2008 that Kougouris teaches creating code for one or more forms including ones of the set of one or more attributes, the claim is not patentably distinct from that affirmed by the Board of Patent Appeals and Interferences. The rejections for claims 1-2 and 4-10 and 37 are sustained.

Conclusion

16. Any inquiry concerning this communication or earlier communications from the examiner should be directed to RACHNA S. DESAI whose telephone number is (571)272-4099. The examiner can normally be reached on M-F (8:30AM-6:00PM).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Doug Hutton can be reached on 571-272-4137. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Rachna S Desai/
Primary Examiner, Art Unit 2176
07/26/10